

# Exploring the potential energy landscape over a large parameter-space

Yang-Hui He,<sup>a,b,c</sup> Dhagash Mehta,<sup>d</sup> Matthew Niemerg,<sup>e</sup> Markus Rummel<sup>f</sup>  
and Alexandru Veleanu<sup>g</sup>

<sup>a</sup>Department of Mathematics, City University,  
London, EC1V 0HB, U.K.

<sup>b</sup>School of Physics, NanKai University,  
Tianjin, 300071, P.R. China

<sup>c</sup>Merton College, University of Oxford,  
Oxford OX14JD, U.K.

<sup>d</sup>Department of Physics, Syracuse University,  
Syracuse, NY 13244, U.S.A.

<sup>e</sup>Department of Mathematics, Colorado State University,  
Fort Collins, CO 80523, U.S.A.

<sup>f</sup>II. Institut für Theoretische Physik der Universität Hamburg,  
D-22761 Hamburg, Germany

<sup>g</sup>Trinity College, University of Oxford,  
Oxford OX14JD, U.K.

E-mail: [hey@maths.ox.ac.uk](mailto:hey@maths.ox.ac.uk), [dbmehta@syr.edu](mailto:dbmehta@syr.edu),  
[niemerg@math.colostate.edu](mailto:niemerg@math.colostate.edu), [markus.rummel@desy.de](mailto:markus.rummel@desy.de),  
[alexandru.valeanu@trinity.ox.ac.uk](mailto:alexandru.valeanu@trinity.ox.ac.uk)

**ABSTRACT:** Solving large polynomial systems with coefficient parameters are ubiquitous and constitute an important class of problems. We demonstrate the computational power of two methods — a symbolic one called the Comprehensive Gröbner basis and a numerical one called coefficient-parameter polynomial continuation — applied to studying both potential energy landscapes and a variety of questions arising from geometry and phenomenology. Particular attention is paid to an example in flux compactification where important physical quantities such as the gravitino and moduli masses and the string coupling can be efficiently extracted.

**KEYWORDS:** Flux compactifications, Differential and Algebraic Geometry, Superstring Vacua

ARXIV EPRINT: [1301.0946](https://arxiv.org/abs/1301.0946)

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Parametric potentials	3
<b>2</b>	<b>Algebraic geometry &amp; the comprehensive Gröbner basis method</b>	<b>5</b>
2.1	The comprehensive Gröbner basis	5
<b>3</b>	<b>Numerical algebraic geometry and coefficient-parameter polynomial continuation</b>	<b>8</b>
3.1	Coefficient-parameter polynomial continuation	9
<b>4</b>	<b>Illustrative examples</b>	<b>12</b>
4.1	<i>Sys1</i> : a single-modulus example	12
4.2	<i>Sys2</i> : a two-moduli model	14
4.3	<i>Sys3</i> : the quintic	16
4.4	Further applications: <i>Sys4</i>	19
<b>5</b>	<b>Conclusion and outlook</b>	<b>20</b>

---

## 1 Introduction

The hypersurface defined by a potential energy function  $V(\vec{x}; \vec{a})$ , with  $\vec{x} = (x_1, \dots, x_N)$  being the variables and  $\vec{a} = (a_1, \dots, a_k)$  being a set of parameters, is called the potential energy landscape (PEL) of a given physical model. The landscape, in particular, refers to the potentially large number of parameters which constitutes a moduli space. The special points of a PEL, defined as the critical (or stationary) points of  $\frac{\partial V(\vec{x})}{\partial x_i} = 0$ , with  $1 \leq i \leq N$ , give crucial information about the physical system depending on the problem at hand.

In recent years, a huge influx of advancements have come through from many areas in theoretical physics and chemistry in understanding the PEL and its relation to various physical and chemical properties. The research areas where the PEL methods have been immensely successful in explaining the underlying physics or chemistry include clusters [1–6], disordered systems and glasses [7, 8], biomolecules, protein folding, string phenomenology [9], and within this area flux compactifications [10–16].

Due to the importance of these critical points of the PELs, finding them in realistic models has been an active research area for quite some time. The stationary equations for any realistic model are invariably non-linear which are known to be extremely difficult to solve in general. Various numerical techniques exist based on the Newton-Raphson method and its sophisticated variants [17–21] where a random initial guess is refined to attain a single solution of the system. However, in all these methods, even after feeding a large number of random initial guesses, one can never be sure of obtaining all the solutions.

While *many* solutions as opposed to *all* solutions may be fine in many applications, often one has to find all the solutions. The problem of not obtaining all the solutions becomes crucial when one needs the information about all the local minima or the global minimum, etc. For example, in string theory models, where in many cases the potential energy landscape is defined by an effective four-dimensional,  $\mathcal{N} = 1$  supergravity scalar potential  $V(K, W)$  given a Kähler potential  $K$ , and a superpotential  $W$ , one has to find all the minima. These minima are called *string vacua* and addressing the plethora of solutions is one of the most important of current theoretical challenges.

Fortunately, in a large class of models, the equations are polynomials, or at least they have polynomial-like non-linearity.<sup>1</sup> Once they are identified as a system of polynomial equations, we can use techniques from algebraic geometry, or more specifically, computational algebraic geometry to solve them by systematically transforming a given system of multivariate polynomial equations to another system and has the same solutions as the original one. The new system of equations is called a *Gröbner basis* (GB), and the algorithm to compute one is called the Buchberger algorithm (BA).

The biggest advantage here is that one can find all the solutions of the given system once the computation is finished. Computational algebraic geometry, which is essentially a set of techniques based on the Gröbner basis method, has become one of the most useful tools to study a number of phenomena in theoretical physics. Recently, the rich interplay between algebraic geometry and theoretical physics, especially in gauge and string theory, has been an active area of research [23]. Activities in these areas have been enhanced with the increased power of computers and the development of algorithms in computational algebraic geometry.

More specifically, a variety of methods have been used to study the moduli space of vacua over the past few years [24, 25] based on symbolic computational algebraic geometry, most of whose sub-methods and sub-algorithms rely on Gröbner basis techniques (cf. [26] for an overview on the method). For convenience, a freely available computational package, **StringVacua**, which is a **Mathematica** package specifically designed for phenomenologists [22, 27, 28] exists. **StringVacua** interfaces with the advanced computational algebraic geometry package **Singular** [29]. Using **StringVacua**, one can extract important information such as the dimension of the vacuum, the number of real roots in the system, stability and supersymmetry of the potential, or the branches of moduli space of vacua, etc., using a regular desktop machine in many circumstances.

However, the GB method is known to suffer from *exponential complexity*, i.e., the computation time and the RAM required by the BA algorithm increases exponentially with the number of variables, equations, degree, and terms in each polynomial; it is usually less efficient for systems with irrational coefficient parameters  $\vec{a}$ ; and it is also highly sequential, i.e., very difficult to parallelize the algorithm and put on a big cluster.

To overcome these shortcomings of the GB methods, a different approach called numerical algebraic geometry (NAG) was recently introduced. Its core algorithm, called *nu-*

---

<sup>1</sup> In the presence of non-perturbative effects where exponential terms contribute, one could still reduce the system into polynomials of Lambert functions, perform standard polynomials manipulations and treat the Lambert functions numerically [22].

*merical polynomial homotopy continuation* (NPHC), guarantees to find all of the solutions unlike other numerical methods such as the Newton-Raphson and its variants. Moreover, unlike the Gröbner basis techniques, the NPHC method is “embarrassingly parallelizable”, and hence one can solve more complicated systems efficiently using computer clusters.

NAG was introduced in particle theory and statistical mechanics areas in ref. [30]. Subsequently, the NPHC method was used to solve systems arising in numerous physical phenomena in lattice field theories [31, 32], statistical physics [33–37], particle phenomenology [38, 39], and string phenomenology [40–43].

## 1.1 Parametric potentials

As mentioned above, in generic physical applications, the potential energy function is defined over a possibly vast parameter-space, where each point  $\vec{a}$  represents a different physical situation. For example, in a statistical mechanics model, the parameters are the disorders [7]. Another example in theoretical chemistry where the models are Morse clusters, the parameters represent the strength of the inter-particle potential [44–47]. As a third example, in lattice field theory the parameters represent different background fields in the models [31, 32, 48, 49]. Another situation is in flux compactifications within string phenomenology where the parameters represent the flux quanta.

The flux quanta are discrete parameters that are given as integrals of  $n$ -forms along  $n$ -cycles in a compact space, see section 4. Their discreteness arises as a Dirac quantization condition. For many compactification manifolds, for instance Calabi-Yau spaces, these cycles exist on the order of 10 to 100 where the flux quanta can be chosen independently. This yields an exponentially large parameter space that is only limited by conditions of conservation of certain charges in the compact space.

The stationary equations for parametric potentials obviously have parameters and if we want to use the GB or NPHC methods directly, we have to specify numerical values of the parameters *before* using the methods. In other words, we have to solve the system separately from scratch for each point in the parameter space. In practice, due to the large number of physically interesting parameter-points, this crude way of solving the problem becomes prohibitively time consuming as well as computationally expensive.

In this paper, we introduce two methods which deal with parametric systems extremely efficiently. The first method is called the *Comprehensive Gröbner basis* (CGB) method. This is a completely symbolic method where given a parametric system of equations, CGB yields a new system, leaving all the parameters in the symbolic form, which is a GB for all values of the parameters and also for all the *specializations*, i.e., special values of the parameters. Once we obtain a CGB for a given system, it then only amounts to inputting the specific values of parameters and extracting the corresponding solutions out. Thus, we can efficiently solve the system for as many parameter-points as we desire since the system is *solved* for the whole parameter space. However, as one may surmise, this method has the same aforementioned short-comings as the usual GB method. Nevertheless, when successful in finishing the computation, the CGB method reduces the amount of work in finding string vacua over a large space of flux parameters drastically.

The second method is called *coefficient-parameter polynomial continuation* (CPPC), which is based on NAG. In NPHC, one first has to estimate an upper bound on the number of solutions of the given system so that one can then construct another system which has the same number of solutions as the estimate. To reduce the computation in this method, we must come up with a tighter upper bound on the number of solutions. Three of the most important solution bounds were demonstrated in our previous works [30, 31, 33, 41]. However, in most of the realistic systems, the sparsity of the systems (i.e., the number of monomials in each equation may be only a few) is not fully taken into account. Thus, even though the original system may have only a few solutions, we may have to track many paths making it computationally expensive. CPPC relies on the fact that the maximum number of solutions of a parametric system of polynomial equations over all the parameter-points is that at a generic parameter-point [50]. The number of paths to be tracked is usually drastically smaller than any of the other upper bounds for the CPPC.

The second, and perhaps the most crucial, advantage is that in the CPPC method, we obtain the start system once for for a generic parametric system, i.e., we then can use the same start system for arbitrary parameter-points in the entire parameter-space. Several more advantages of this method exist when solving these systems on a computer cluster, but most importantly the method is “embarrassingly parallelizable”, a subject matter to which we shall later return.

A very efficient package, which is yet to be publicly available, called *Paramotopy* [51], uses all the computational advantages of the method in our favor and can deal with hundreds of thousands of parameter-points. In this paper, we heavily rely on *Paramotopy* for our computations.

We would like to compute solutions as a function of the parameters. We can do this, although, implicitly, with the use of the *Paramotopy* package. *Paramotopy* allows the user to define the parameter points using a Monte-Carlo style file, and so a user can impose the necessary relationship on the parameters. In addition, one could treat all the parameters (or just some of them) as variables when necessary and use the conventional polynomial continuation methods. One then adds appropriate equations that correspond to the real values of the parameter points. Alternatively, one can also do the same with GB methods. However, in both situations, introducing more variables decreases the speed in these calculations. Hence, the *Paramotopy* package provides a much quicker method for computing solutions at multiple parameter values of interest.

We would like to make the clarification that the above two methods are based on complex algebraic geometry which means that the variables are first considered to be complex, and then once the solutions are obtained, only purely real solutions are retained for analysis if the system was supposed to be made of real variables. However, there is a more recent method from real algebraic geometry, based on the discriminant varieties [52, 53], which treats both variables and parameters as reals from the beginning and gives a different set of information than the methods described in this paper. The details on this method with applications arising from physics will be addressed in forthcoming works.

We first introduce the Comprehensive Gröbner basis method and solve a few toy models in section 2. Then, to make the paper self-contained, we briefly explain the NPHC method

before explaining the coefficient-parameter polynomial continuation that is implemented in section 3. Therein, we also introduce the salient features of the Paramotopy package. In section 4, we solve several toy models as well as more realistic models such as flux compactifications on the quintic manifold. We will extract some interesting physics using the string vacua of these models over a large space of fluxes. Finally, we conclude with remarks and prospects in section 5.

## 2 Algebraic geometry & the comprehensive Gröbner basis method

In this section, we explain the concept of a Comprehensive Gröbner basis in more detail and some basics of algebraic geometry to set the nomenclature and also to facilitate the reader. We first introduce few technical terms leading to the definition of a Gröbner basis. Then, we will explain what a Comprehensive Gröbner basis is. Readers uninterested in the technicalities involved in the definition may freely skip subsection section 2.1 after reading the next two paragraphs.

Roughly speaking, given a system of polynomial equations, the Buchberger Algorithm (BA), or its refined variants, compute a new equivalent system of polynomials, called a Gröbner basis [54] which has nicer properties; using the BA on multivariate polynomial systems is analogous to Gaussian elimination for linear systems. Nowadays, efficient variants of the BA are available, e.g., F4 [55], F5 [56], and Involution Algorithms [57]. Symbolic computation packages such as Mathematica, Maple, Reduce, etc., have built-in commands to calculate a Gröbner basis. Moreover, Singular [29], COCOA [58], and Macaulay2 [59] are specialized packages for computational algebraic geometry available freely. MAGMA [60] is also such a specialized package available commercially.

A system of polynomial equations with parameters is called a parametric system; finding the critical points of a polynomial PEL is precisely such a system. If one is interested in solving the system at finitely many points on the parameter-space, then inserting the numerical values of the parameters in the system and obtaining a Gröbner basis is a quick escape, especially for a small number of parameters. A natural question to ask is if it is possible to obtain a Gröbner basis for a given monomial ordering in terms of the symbolic form of the parametric coefficients, valid for all its special cases, called *specializations*. One can indeed compute such a ‘parametric Gröbner basis’ called Comprehensive Gröbner basis (CGB) [61]. Algorithmically, we use the internal libraries of Singular to compute the CGB in this paper.

### 2.1 The comprehensive Gröbner basis

The technicalities in this subsection will lead to a very useful result, namely that we can transform a given system of multivariate polynomial equations to another one which has the same solutions but is easier to solve. Here, the original system is considered as a basis of an algebraic object, called an *ideal*. Then an important result, that an appropriate change of this basis leaves the solution space unchanged, is used.

**Polynomial rings.** We define a polynomial  $f$  as

$$f = \sum_{\alpha} a_{\alpha} x^{\alpha}. \tag{2.1}$$

Here, the sum is over a finite number of  $m$ -tuples  $\alpha = (\alpha_1, \dots, \alpha_m)$  and  $x^{\alpha} = x_1^{\alpha_1} \dots x_m^{\alpha_m}$  is a monomial with all  $\alpha_i$  being non-negative integers. The coefficients  $a_{\alpha}$  and the variables  $x_i$  take values from the field  $K$ . However, to use the results of Algebraic Geometry to its full extent, unless otherwise specified, we will take  $K = \mathbb{C}$ .

Now, if  $K[x_1, \dots, x_m]$  is the set of all polynomials in variables  $x_1, \dots, x_m$  with coefficients in  $K$ , then  $f \in K[x_1, \dots, x_m]$  can be viewed as a function  $f : K^m \rightarrow K$  where  $K^m$  is the affine space of all coefficients. Thus, the sum and product of two polynomials is a polynomial, and a polynomial  $f$  divides a polynomial  $g$  if and only if  $g = fh$ , for some  $h \in K[x_1, \dots, x_m]$ . Using this, it can be shown that under addition and multiplication,  $K[x_1, \dots, x_m]$  satisfies all of the field axioms except for the existence of a multiplicative inverse because  $\frac{1}{x}$  is not a polynomial. Indeed,  $K[x_1, \dots, x_m]$  satisfies the axioms for a commutative ring, or more precisely a *polynomial ring*<sup>2</sup>.

**Ideal.** One can now view all the polynomials of a system of polynomial equations as *elements* of a polynomial ring. Hence, one can also define a corresponding *vector space*, called an ideal. More specifically, an ideal  $I$  is a subset of  $K[x_1, \dots, x_m]$  with the following properties:

1.  $0 \in I$ ,
2.  $f + g \in I$  for all  $f, g \in I$ , and
3.  $hf \in I$  for  $f \in I$  and  $h \in K[x_1, \dots, x_m]$ .

Consider any  $f \in I \subset K[x_1, \dots, x_m]$ . If  $f$  can be written as  $f = \sum_{\alpha} a_{\alpha} h_{\alpha}$  with  $a_{\alpha} \in K$  and  $h_{\alpha} \in K[x_1, \dots, x_m]$ , then we write  $I = \langle h_{\alpha} \rangle \subset K[x_1, \dots, x_m]$ . If the indexing set  $\alpha$  is finite, say with cardinality  $t$ , then  $I$  is called a *finitely generated ideal*. The polynomials  $h_1, \dots, h_t$  are then said to be a *finite basis* of  $I$ , and we write  $I = \langle h_1, \dots, h_t \rangle$ .

**Affine variety.** So far, we have introduced the algebraic counterpart of Algebraic Geometry. The solution space of a given ideal is called a *variety*. Specifically, an affine variety of an ideal  $I = \langle h_1, \dots, h_t \rangle$  is the set of common zeros of polynomials  $h_1, \dots, h_t$  in affine space, denoted as  $V(h_1, \dots, h_t)$  or  $V(I)$ .

**Gröbner basis.** The formalism of Algebraic Geometry turns out to be very helpful. Interpreting the polynomials  $h_i$  as a basis of  $I$ , we can *change the basis* to, say,  $\langle H_1, \dots, H_s \rangle$ . Then, it can be shown that the solution space remains unchanged in an appropriate change of basis, that is,  $V(h_1, \dots, h_t) = V(H_1, \dots, H_s)$ . In essence, we use computational techniques to find a basis that is easier to deal with than the original one, in a certain sense. Such a basis is called a Gröbner basis.

---

<sup>2</sup>For a nice discussion on related topics, the reader is referred to [54, 62].

In linear algebra, such a change of basis can be done via Gaussian Elimination, and the new basis is the familiar Row-Echelon form. In general, an algorithm to obtain a Gröbner basis performs a specific set of algebraic operations including factorizing and dividing the polynomials. In any algorithm that computes a Gröbner basis, the division requires one to impose a *total order* among the monomials. This is called a *monomial ordering*.

A monomial ordering is a relation ‘ $\succ$ ’ on the set of monomials  $x^\alpha$ ,  $\alpha \in \mathbb{Z}_{\geq 0}^n$ , satisfying the following properties:

1. the ordering always tells which of two distinct monomials is greater,
2. the relative order of two monomials does not change when they are each multiplied by the same monomial, and
3. every strictly decreasing sequence of monomials eventually terminates [63].

Different types of monomial orderings exist that satisfy the aforementioned properties such as lexicographic, graded lexicographic, graded reverse lexicographic, or degree lexicographic. Different monomial orderings are useful depending on the algorithm that is employed to compute the Gröbner Basis. Lexicographic orderings will be primarily used throughout our discussion. To learn more about monomial ordering, the reader is referred to [54, 62].

By fixing a monomial order, we define a *leading term* for each polynomial of a given ideal, denoted as  $\langle LT(h_1), \dots, LT(h_t) \rangle$ . One can always find a finite subset  $G = \langle H_1, \dots, H_s \rangle$  of an ideal  $I$  (except for the trivial case  $I = \langle 0 \rangle$ ) such that every leading term of  $f \in I$  can be generated by  $\langle LT(H_1), \dots, LT(H_s) \rangle$ . Here,  $f \in I$  means that  $f$  is an algebraic combination of  $h_1, \dots, h_t$ , as is required for  $I$  to be an ideal. Such a subset  $G$  is called a Gröbner basis with respect to the specific monomial order.<sup>3</sup>

One can show that for any given monomial order, every nontrivial ideal  $I \subset K[x_1, \dots, x_m]$ , has a Gröbner basis and that any Gröbner basis for an ideal  $I$  is a basis of  $I$ . One can also show that  $V(I)$  can be computed by any basis of  $I$ , and so the solutions of  $I$  are the same as that of any of its Gröbner basis for any monomial ordering.

A well-defined procedure exists to compute a Gröbner basis for any given ideal and monomial ordering, called the Buchberger algorithm. It should be noted that the Buchberger algorithm reduces to Gaussian elimination in the case of linear equations, as it is a generalization of the latter. Similarly, it is a generalization of the Euclidean algorithm for the computation of the Greatest Common Divisors of a univariate polynomial.

**Comprehensive Gröbner basis.** If the leading coefficient of each element of the basis is 1 and no monomial in any element of the basis is in the ideal generated by the leading terms of the other elements of the basis, the basis is called a reduced Gröbner basis. A reduced Gröbner basis is unique for a given ideal and monomial ordering, unlike a Gröbner basis.

---

<sup>3</sup>It should be noted here that a Gröbner basis may not be unique for a fixed monomial ordering. So, we call it *a* Gröbner basis rather than *the* Gröbner basis. However, the so-called reduced Gröbner basis is unique for a given monomial ordering. The reader is referred to ref. [54, 62] for more details.



Now, if we have a parametric ideal, i.e.,  $I = \langle f_1, \dots, f_s \rangle \subset R[x_1, \dots, x_n; a_1, \dots, a_m]$ , where  $R$  is a unique factorization domain, then a Comprehensive Gröbner basis (CGB) is the distinct reduced Gröbner basis for all possible values of the parameters  $a_1, \dots, a_m$  [61]. There are several algorithms available to compute the CGB [64–66]. We refer the reader willing to learn more about the actual algorithm and related issues to these references, and leave the section by noting that we use the internal libraries of Singular to compute the CGB in this paper.

Let us illustrate with a trivial example of a nonlinear parametric equation  $ax^2 + bx + c = 0$ ; this equation defines an ideal in  $\mathbb{C}[x]$  with parameters  $a, b, c$  in one variable  $x$ . Singular’s CGB library yields that there are three cases:

1. Case-1: for  $a = b = c = 0$ , the solution is the whole  $\mathbb{C}$ , i.e. all  $x \in \mathbb{C}$ ;
2. Case-2: for  $a = 0$  and  $b \neq 0$ , the solution is the line  $bx + c = 0$ ; and
3. Case-3: for  $a \neq 0$ , the solution is the quadric  $ax^2 + bx + c = 0$ ,

as is clearly expected.

Next, let us consider a more involved example. Take the bi-variate system of two equations  $\langle f_1 = ax^2y^2 + bxy + 2 = 0, f_2 = bx + ay + 2 = 0 \rangle$ , where  $a$  and  $b$  are parameters and  $x$  and  $y$  are variables. Fix the lexicographic ordering  $x \succ y$ . Then, the leading terms are clearly  $LT(f_1) = ax^2y^2$  and  $LT(f_2) = bx$ . The Comprehensive Gröbner basis is as follows:

1. Case-1: for  $a = b = 0$ , the solution set empty;
2. Case-2:  $a \neq 0, b = 0$ , the solution space is given by  $ay + 2 = 0 = -2x^2 - a$ ;
3. Case-3:  $a = 0, b \neq 0$ , the solution space is given by  $-y + 1 = 0 = bx + 2$ ;
4. Case-4:  $ab \neq 0$ , the solution space is given by  $-a^3y^4 - 4a^2y^3 + (b^2 - 4)ay^2 + 2b^2y - 2b^2 = 0$  and  $bx + ay + 2 = 0$ ;

the last three cases can, of course, be checked by simple substitution.

### 3 Numerical algebraic geometry and coefficient-parameter polynomial continuation

Having expounded on the virtues of the CGB, we now introduce a parallel method, which attacks our problem from an entirely different perspective. The numerical polynomial homotopy continuation (NPHC) method [63] is a recently introduced numerical method that finds all the solutions of the given system of polynomial equations. It has been used in various problems in particle theory and statistical mechanics in refs. [30–36, 38, 40, 41]. Here we briefly explain the NPHC method to make the paper self-contained.

For a system of polynomial equations,  $P(x) = 0$ , where  $P(x) = (p_1(x), \dots, p_m(x))^T$  and  $x = (x_1, \dots, x_m)^T$ , which is known to have isolated solutions, the Classical Bézout Theorem asserts that for generic values of coefficients, the maximum number of isolated

solutions in  $\mathbb{C}^m$  is  $\prod_{i=1}^m d_i$ , where  $d_i$  is the degree of the  $i$ th polynomial. This bound, the *classical Bézout bound* (CBB), is exact for generic values. The *genericity* is well-defined and the interested reader is referred to ref. [63, 67] for details.

Based on the CBB, a *homotopy*  $H(x, t)$  can be constructed as

$$H(x, t) = \gamma(1 - t)Q(x) + t P(x), \tag{3.1}$$

where  $\gamma$  is a generic complex number and  $Q(x) = (q_1(x), \dots, q_m(x))^T$  is a system of polynomial equations with the following properties:

1. the solutions of  $Q(x) = H(x, 0) = 0$  are known or can be easily obtained.  $Q(x)$  is called the *start system* and the solutions are called the *start solutions*;
2. the number of solutions of  $Q(x) = H(x, 0) = 0$  is equal to the CBB;
3. the solution set of  $H(x, t) = 0$  for  $0 \leq t \leq 1$  consists of a finite number of smooth paths, called homotopy paths, each parameterized by  $t \in [0, 1)$ ; and
4. every isolated solution of  $H(x, 1) = P(x) = 0$  can be reached by some path originating at a solution of  $H(x, 0) = Q(x) = 0$ .

We can then track all the paths corresponding to each solution of  $Q(x) = 0$  from  $t = 0$  to  $t = 1$ . The paths which reach  $P(x) = 0 = H(x, 1)$  are the solutions of  $P(x) = 0$ . By implementing an efficient path tracker algorithm, all isolated solutions of a system of multivariate polynomials system can be obtained: it is shown [63] that for a generic  $\gamma$ , there are no singularities (i.e., paths do not cross each other) for  $t \in [0, 1)$ . Thus, in the end, we obtain all the solutions of the system  $P(x) = 0$ . In this respect, the NPHC method has a great advantage over all other known methods for finding stationary points.

Several sophisticated numerical packages well-equipped with path trackers exist, such as Bertini [68], PHCpack [69], HOMPACK [70], and HOM4PS2 [71, 72], which are all available as freewares from the respective research groups.

### 3.1 Coefficient-parameter polynomial continuation

The advantages of the homotopy based on the CBB are (1) the CBB is easy to compute, and (2) the start system based on the CBB can be solved quickly. The drawback of it is that the CBB does not take the sparsity of the system into account; systems arising in practice have far fewer solutions than the CBB, so a large portion of the computational effort is wasted.

Hence, one can also use homotopies based on tighter upper bounds. For example, the 2-Homogeneous Homotopy is constructed by first writing  $\mathbb{C}^m = \mathbb{C}^k \times \mathbb{C}^{k-m}$  for some  $k$  where  $0 < k < m$ , which is accomplished by partitioning the original variables into two groups. This has the advantage of incorporating some of the structure of the given polynomial system  $P(x)$  into the start system  $Q(x)$ . The corresponding bound, called the 2-Homogeneous Bézout Bound (2HomBB), is often tighter than the CBB when the polynomial system  $P(x)$  has a naturally arising partition of the variables, which occurs

in the examples below. Given a partition, the 2HomBB is easy to compute and the start system can be solved quickly via linear algebra.

Another important homotopy is the Polyhedral Homotopy which uses the monomial structure of the given polynomial system  $P(x)$  based on the Bernstein-Khovanskii-Kushnirenko (BKK) Theorem [73–75] to yield the BKK bound. Essentially, this upper bound on the number of complex solutions is obtained by computing the mixed volume of the convex hull of the Newton polytope (which is based on the exponents of the monomials appearing) of each equation. The interested reader from the physics community is referred to ref. [30, 41] for these above two bounds. We note that, as with the CBB, the 2HomBB and BKK bounds are also generically sharp with respect to the family of polynomial systems under consideration.

However, in the realistic systems, we do need to take the sparsity of the systems fully into account. Indeed, even though the original system may have only a few solutions, we may have to track many paths making it computationally expensive. The *coefficient-parameter polynomial continuation* method is a much more practical way to overcome this difficulty. The crux of using this method relies on a theorem, which we state without its proof below, that states that the maximum number of solutions of a parametric system of polynomial equations over all the parameter-points is the one at a generic parameter-point [50, 76]:

**Theorem 1.** *Let  $P(x, \lambda) = 0$  be a system of polynomial equations,  $p_1(x, \lambda), \dots, p_n(x, \lambda) = 0$ , where  $\lambda = (\lambda_1, \dots, \lambda_m) \in \mathbb{C}^m$  are parameters and  $x = (x_1, \dots, x_n) \in \mathbb{C}^n$  be variables. Then, there exists an open, dense, full-measure set  $U \subset \mathbb{C}^{n+m}$  such that for  $(b_1^*, \dots, b_n^*, \lambda_1^*, \dots, \lambda_m^*) \in U$  the following holds:*

1. *The set  $X^*$  of solutions  $x = (x_1, \dots, x_n)$  of the system*

$$p_1(x_1, \dots, x_n, \lambda_1^*, \dots, \lambda_m^*) + b_1^* = 0,$$

...

$$p_n(x_1, \dots, x_n, \lambda_1^*, \dots, \lambda_m^*) + b_n^* = 0$$

*consists of  $d_0$  isolated points for  $d_0 \leq d$ , where  $d$  is the total degree of the system for a generic  $\lambda$ .*

2. *Smoothness and accessibility properties (Properties 3 and 4 of homotopies) still hold for coefficient-parameter polynomial continuation which is given as follows:*

$$H(x, t) = P(x_1, \dots, x_n, (1-t)\lambda_1^* + t_1, \dots, (1-t)\lambda_m^* + t\lambda_m) + (1-t)b^*$$

*where  $b^* = (b_1^*, \dots, b_n^*)$ . It follows that every solution of  $P(x) = 0$  is reached by a path beginning at a point of  $X^*$ .*

Another way of viewing this is to say that a “special” choice of our coefficients may cause the system to be deficient in the maximum number of solutions. If we let  $D$  be the set of all  $\lambda$  that cause these deficiencies, then  $D$  is a set of measure 0. Hence, CPPC relies on the fact that we choose generic, or random, values of the coefficients. If we choose random values for  $\lambda^*$ , then with probability 1,  $\lambda^* \notin D$ .

Algorithmically, for a given parametric system  $P(x, \lambda) = 0$  where  $\lambda = (\lambda_1, \dots, \lambda_m)$  are parameters, we first simply need to solve  $P(x, \lambda)$  at a generic parameter-point  $\lambda^* \in \mathbb{C}^m$ . This part has to be solved using a homotopy based on total degree, 2-homogeneous, or the BKK root counts. Then, in the second part, the system  $P(x, \lambda^*) = 0$  becomes the start system of all other parameter-points  $\lambda \in \mathbb{C}^m - \{\lambda^*\}$  and the solutions of this system become the start solutions. Finally, each path is tracked with the below homotopy:

$$H(x, \lambda, t) = (1 - t)P(x, \lambda^*) + t P(x, \lambda). \tag{3.2}$$

The most important trick here is to choose the generic parameter-point from the complex space. Once the generic parameter-point is chosen from the complex space, the other parameter point can be chosen to be real if the physical situation requires. Also note that the gamma trick is implicitly employed since the  $\lambda^*$  are generic.

The advantage of coefficient-parameter polynomial continuation over the usual NPHC is huge. When using CPPC, the number of start solutions is drastically smaller than any of the other upper bounds usually (in the worst case, it is equal to the smallest of all the other upper bounds), and hence the number of paths to be tracked reduces a lot. A crucial advantage is that in coefficient-parameter polynomial continuation, we can obtain the start system for a generic parametric system once during a computationally expensive ‘offline’ run. This allows us to use the same start system for any number of parameter-points in the parameter-space and compute the solutions at each parameter point of interest with a much faster ‘online’ run.

Moreover, the system at each parameter-point can be solved completely independent from any other parameter-point. In addition, while solving the system at each parameter-point, each path can also be tracked independently of all others, making it “doubly parallelizable”. We should emphasize that the packages like Bertini and PHCpack now have an implementation of coefficient-parameter polynomial continuation in addition to the usual NPHC method.

A recently developed software module of Bertini, called Paramotopy [51], is a different implementation of the coefficient-parameter polynomial continuation method than the standard one already provided with Bertini and can deal with a huge number of parameter points in parallel. In this paper we have extensively used this software. Several salient features of this new software are:

1. It offers a few options on choosing the parameter-points: the user can demand the package to discretize the parameter-space within a given range in as many parameter-points as required or the package can also be fed in a list of parameter-points provided by the user.

2. Paramotopy first stores the data in the RAM before writing to the hard-disk which is an efficient data-management practice making the package more efficient while dealing with many parameter-points simultaneously.
3. The package is able to disregard any information other than the type of solutions the user requires in the end, i.e., it can be asked to save only real affine solutions, or only non-singular solutions, etc. This is an important aspect of the package because by not including all the information produced in a regular run, it eliminates massive data proliferation.
4. If, for any parameter-point where certain paths require higher precision, Paramotopy informs us so that the user can re-run these specific parameter-points at higher precision settings. Note that the parameter-points may lie on some algebraic subset in which exists an algebraic relationship on the parameters. In this case, the number of complex solutions shrinks, and hence the related paths may require higher precision.

## 4 Illustrative examples

Having explained our methods in detail, in this section, we will illustrate with examples coming from a variety of physical situations, commencing with a toy model and moving onto more serious and involved cases.

### 4.1 *Sys1*: a single-modulus example

We begin with a single-modulus toy example. First, we recall that given the Kähler potential  $K$  and superpotential  $W$ , both as polynomials in fields  $\phi_{A=1,\dots,n}$  one can proceed to construct the scalar potential from the standard formulae [77]:

$$V = e^K \left[ \mathcal{K}^{A\bar{B}} D_A W D_{\bar{B}} \bar{W} - 3|W|^2 \right] . \tag{4.1}$$

As usual the  $D_A$  represents the Kähler derivative  $\partial_A + \partial_A(K)$ , and  $\mathcal{K}^{A\bar{B}}$  is the inverse of the field space metric

$$\mathcal{K}_{A\bar{B}} = \partial_A \partial_{\bar{B}} K . \tag{4.2}$$

Our example of this problem requires the solution of the critical set

$$\partial_A V = 0, \text{ for } A = 1, \dots, n . \tag{4.3}$$

We can further classify the solutions to (4.3) by the amount of supersymmetry they preserve, the value of the bare cosmological constant they dictate, and so forth. The most relevant examples are:

- SUSY, Minkowski:  $\partial_A V = 0, D_A W = 0$  for all  $A$  and  $W = 0$ ;
- SUSY, AdS:  $\partial_A V = 0, D_A W = 0$  for all  $A$  but  $W \neq 0$ .

Thus prepared, let us take a single field example, which is addressed in the demo of `StringVacua` and with which we can compare. Let the Kähler potential  $K$  and superpotential  $W$  of an  $\mathcal{N} = 1$  supersymmetric theory in a single complex moduli field  $T$  be given as

$$K = -3 \log(T + \bar{T}), \quad W = a + bT^8. \quad (4.4)$$

Note that the field  $T$  comes along with its complex conjugate. Even though they can be treated as different variables by merely relabeling them, they are not actually independent variables. To avoid this problem, we can write them in terms of real and imaginary parts, i.e.,  $T = t + i\tau$  where  $t$  and  $\tau$  are real. The potential, using (4.1), is

$$V = \frac{1}{3t} (4b(5b(t^2 + \tau^2)^7 - 3a(t^6 - 21t^4\tau^2 + 35t^2\tau^4 - 7\tau^6))) \quad (4.5)$$

which has 2 variables. To find the stationary points of  $V$ , we need to compute the zero locus of the partial derivatives of  $V$  with respect to variables  $t$  and  $\tau$ :

$$\begin{aligned} \frac{\partial V}{\partial t} &= \frac{1}{3t^2} (4b(5b(13t^2 - \tau^2)(t^2 + \tau^2)^6 - 3a(5t^6 - 63t^4\tau^2 + 35t^2\tau^4 + 7\tau^6))) = 0, \\ \frac{\partial V}{\partial \tau} &= \frac{1}{3t} (56b\tau(5b(t^2 + \tau^2)^6 + a(9t^4 - 30t^2\tau^2 + 9\tau^4))) = 0. \end{aligned} \quad (4.6)$$

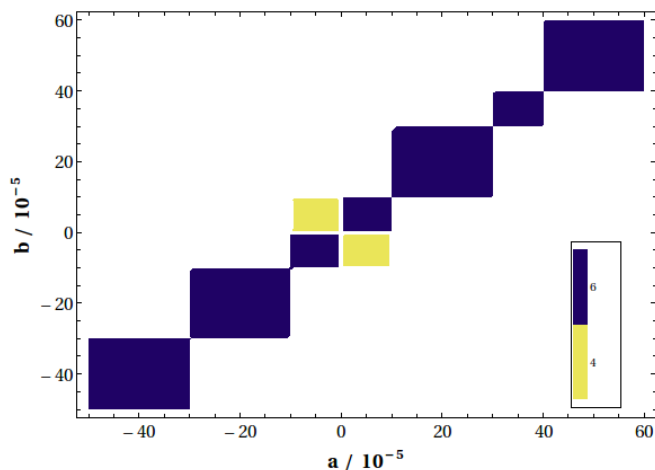
For general values of parameters  $a$  and  $b$  the system already becomes difficult to analyze using symbolic methods and one could solve the system for specific values such as  $a = b = 1$  [26, 27].

We also note that the stationary equations in this example involve denominators. Since we are not interested in the solutions for which the denominators are zero, we clear them out by multiplying them with the numerators appropriately. In these equations, all the denominators are multiples of  $t$ . The condition that none of the denominators is zero can be imposed algebraically by adding an additional equation,  $1 - yt = 0$ , with  $y$  being an additional variable. Thus, we now have 3 equations in 3 variables. The CGB library of `Singular` can deal with this system. The expression of the CGB of this system is quite large so we do not write it down here.

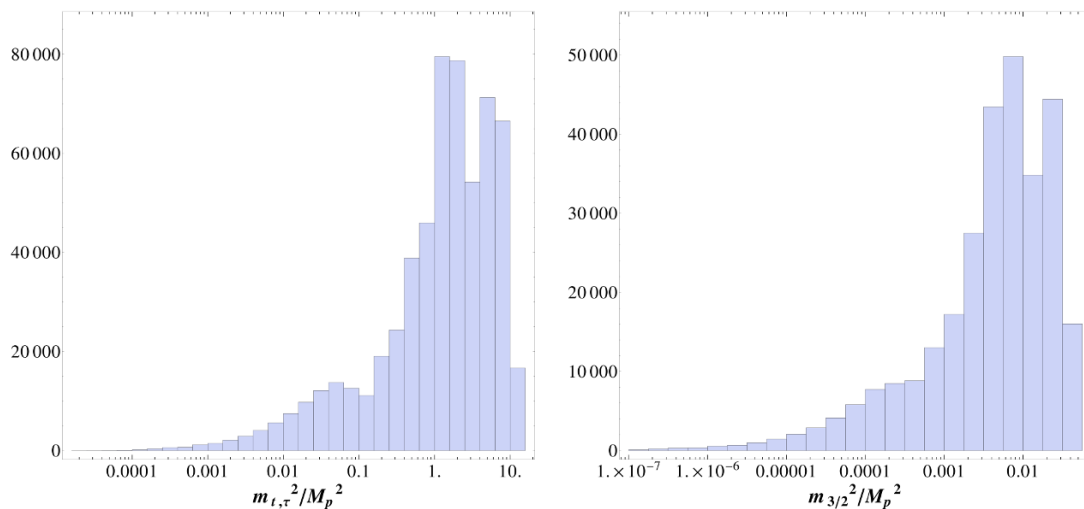
We can also solve this system using CPPC. For the range of values of  $a$  and  $b$  given in figure 1 we find that there are either 6 solutions for  $a, b > 0$  and  $a, b < 0$  or 4 solutions for  $a < 0, b > 0$  and  $a > 0, b < 0$ ; these are indicated by different colors in figure 1. We scan over a total of 100,150 parameter points and find a total of 582,676 solutions. Exactly half of the solutions are physical, i.e.  $t > 0$  which corresponds to a positive volume of the cycle associated to the Kähler modulus  $T$ . We did our computation on a desktop machine on single processor (Linux machine with 2.1GHz clock speed). First, Bertini takes around 30 minutes to solve the system from scratch for a given parameter-point. This means that it would have taken around 5.731 years to solve the system at all the 100,150 parameter-points. With the CH method, however, we solved them in only 55 hours.

The scalar masses of the moduli  $t$  and  $\tau$  which can be calculated as the eigenvalues of the Hessian of  $V$  are found to be positive for all parameter points with  $t > 0$ . We give the moduli masses and the gravitino mass

$$m_{3/2}^2 = e^K |W|^2, \quad (4.7)$$



**Figure 1.** Scanned values for  $a$  and  $b$  for the one-modulus example in *Sys1*. The corresponding number of solutions of the system per parameter point is indicated by the color. The total number of parameter points is 100,150 and the spacing between the points is equidistant.



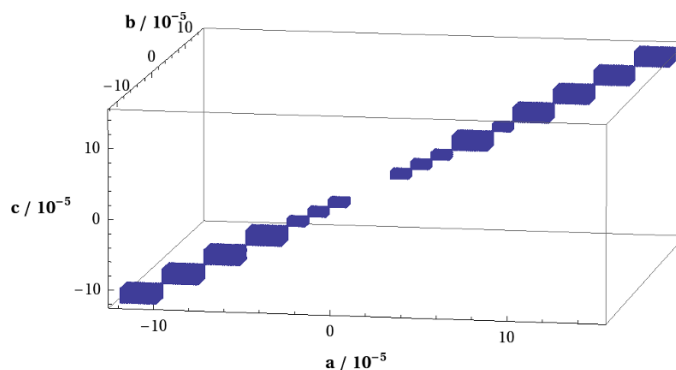
**Figure 2.** Masses of the moduli  $t$  and  $\tau$  as eigenvalues of the Hessian (left) and the gravitino mass (right) for the one-modulus model *Sys1*.

which determines the scale of supersymmetry breaking in figure 2, which is a frequency plot of the mass values for our space of 100,150 parameters.

#### 4.2 *Sys2*: a two-moduli model

Now, let us move on to a model with two moduli fields. Consider the Kähler potential and superpotential

$$\begin{aligned}
 K &= -3 \log(T + \bar{T}) - \log(S + \bar{S}), \\
 W &= aS + bST + cT^2,
 \end{aligned}
 \tag{4.8}$$



**Figure 3.** Scanned values for  $a$  and  $b$  and  $c$  for the two-moduli example *Sys2*. The total number of parameter points is 100,672, and the spacing between the points is equidistant. The blue regions indicate that there are 6 solutions to *Sys2*.

with two fields, a Kähler modulus  $T = t + i\tau$  and the axio-dilaton  $S = s + i\sigma$ . Hence, using (4.1), the stationary equations to be solved are:

$$\begin{aligned}
 0 &= (3a^2(s - \sigma)(s + \sigma) + 6at(b(-s^2 + \sigma^2) + 2c\sigma\tau) + 2bc\sigma\tau(t^2 - 3\tau^2) \\
 &\quad - b^2(s - \sigma)(s + \sigma)(5t^2 - 3\tau^2) + c^2(5t^2 - 3\tau^2)(t^2 + \tau^2)), \\
 0 &= (-9a^2(s^2 + \sigma^2) + b^2(s^2 + \sigma^2)(5t^2 - 9\tau^2) + 2bc\tau(\sigma t^2 + 18st\tau - 9\sigma\tau^2) \\
 &\quad + c^2(-5t^4 + 2t^2\tau^2 - 9\tau^4) + 6a(2b(s^2 + \sigma^2)t + c(5st^2 + 4\sigma t\tau - 3s\tau^2))), \\
 0 &= (-c\sigma t(6a + bt)) + (6acs + 3b^2(s^2 + \sigma^2) - 18bcst - 2c^2t^2)\tau + 9bc\sigma\tau^2 + 6c^2\tau^3), \\
 0 &= (3a^2\sigma - 6at(b\sigma + c\tau) + b(-5b\sigma t^2 - ct^2\tau + 3b\sigma\tau^2 + 3c\tau^3)), \\
 0 &= 1 - zts.
 \end{aligned}$$

The first four equations arise from setting the numerators of the various partial derivatives of  $V$  to zero, and the last is an auxiliary equation to ensure that the denominator does not vanish. Therefore, this system is 5 equations in 5 variables with 3 parameters. Again, this is an example used in *StringVacua*, but  $a, b$  and  $c$  are parameters that were chosen to be 1, -1, 1 respectively in [26, 27]. Now, we can undertake the much more challenging task of taking a huge range of parameter-values in our computation.

Again, for generic choice of the values for  $a, b$  and  $c$ , it becomes difficult for the traditional GB method. However, we can compute the CGB for this system; The output from *Singular* of the CGB for this system is quite large, and, hence, we do not write it here.

Using *Paramotopy* on the other hand, we easily scan over 100,672 parameter points, see figure 3. We find 6 solutions per parameter point which yields a total of 604,032 solutions. The physicality condition for the case of this model demands again  $t > 0$  and also  $s = g_s^{-1} > 0$ , i.e. positive string coupling. This is fulfilled by 503,299 solutions. As for *Sys1*, we did our computation on a desktop machine. *Bertini* takes around 40 minutes to solve the system from scratch for a given parameter-point. Hence, for all the 100,672 parameter-points, it would have taken 7.682 years. However, using the *CPPC* method, we solved them all in 125 hours. When evaluating the Hessian of  $V$ , we find that there is always at least one negative eigenvalue, i.e. no vacua exist for this model for the set of parameters we considered.



### 4.3 *Sys3*: the quintic

An area of string phenomenology where one can make use of the power of CPPC is the landscape of flux vacua in type IIB string theory [12, 13, 15, 16]. As mentioned in *Sys1*, if one goes to a particular corner of the moduli space, the equations

$$D_i W = 0 \tag{4.9}$$

that determine the supersymmetric vacuum state of the no-scale scalar potential [78, 79]

$$V = e^K K^{a\bar{b}} D_a W \overline{D_b W}, \tag{4.10}$$

are polynomial equations in the complex structure moduli fields  $\phi_a = \tau, U_1, \dots, U_{h^{2,1}}$ , where  $\tau = \sigma + i s$  ( $= i\bar{S}$ ) is the axio-dilaton. The parameters of these equations are the flux integers

$$\begin{aligned} \frac{1}{(2\pi)^2 \alpha'} \int_{A_a} F_3 &= f_{1a} \in \mathbb{Z}, & \frac{1}{(2\pi)^2 \alpha'} \int_{B^a} F_3 &= f_{2a} \in \mathbb{Z}, \\ \frac{1}{(2\pi)^2 \alpha'} \int_{A_a} H_3 &= h_{1a} \in \mathbb{Z}, & \frac{1}{(2\pi)^2 \alpha'} \int_{B^a} H_3 &= h_{2a} \in \mathbb{Z}, \end{aligned} \tag{4.11}$$

where  $F_3$  and  $H_3$  are the RR and NS three-form flux of type IIB string theory, and  $\langle A_a, B^b \rangle$  is a symplectic basis for the  $b_3 = 2h^{2,1} + 2$  three-cycles. The flux integers are to be chosen freely as long as the D3 tadpole constraint

$$L = \frac{1}{(2\pi)^4 (\alpha')^2} \int_{X_3} H_3 \wedge F_3 = h_1 f_2 - h_2 f_1 \tag{4.12}$$

is not violated.

For supersymmetric flux configurations,  $F_3$  and  $H_3$  always combine into an imaginary-self-dual (ISD) flux  $G_3 = F_3 - \tau H_3$  [12, 80], which can be written as [9]

$$*_6 s H_3 = -(F_3 - \sigma H_3). \tag{4.13}$$

Consequently, one only has to consider  $2h^{2,1} + 2$  independent directions of the original  $4h^{2,1} + 4$  flux integers defining

$$H_3 = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \quad \text{and} \quad F_3 = \begin{pmatrix} -h_2 \\ h_1 \end{pmatrix}, \tag{4.14}$$

Then, the D3 tadpole eq. (4.12) manifests as a positive definite form, i.e.

$$L = h_1^2 + h_2^2, \tag{4.15}$$

for a symplectic basis of three-cycles. With this form of the D3 tadpole, we can use Paramotopy to find all flux vacua, i.e. solutions to eq. (4.9) for all flux configurations  $h_1, h_2$  with D3 tadpole  $L < L_{\max}$  for a given maximal tadpole  $L_{\max}$ . We have to take into account the  $\text{SL}(2, \mathbb{Z})$  invariance of IIB string theory in order to consider only physically

equivalent flux configurations. It was shown in [81] that for fluxes of the form (4.14), only the configurations

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \cong \begin{pmatrix} -h_1 \\ -h_2 \end{pmatrix} \cong \begin{pmatrix} -h_2 \\ h_1 \end{pmatrix} \cong \begin{pmatrix} h_2 \\ -h_1 \end{pmatrix}. \quad (4.16)$$

are related by  $SL(2, \mathbb{Z})$ , and hence physically equivalent.

This problem to find all supersymmetric flux vacua in the large complex structure limit was carried out for the compactification manifold  $\mathbb{CP}_{11169}^4$  [18] in [81] in the context of de Sitter model building in the Kähler uplifting scenario [82, 83]. To present a simple example, we consider the one complex structure modulus  $\psi$  of the mirror quintic which has  $h^{1,1} = 101$  and  $h^{2,1} = 1$ . In the limit of large complex structure, this describes the one moduli subspace of the quintic [84] given by the vanishing of the polynomial

$$x_0^5 + x_1^5 + x_2^5 + x_3^5 + x_4^5 - 5\psi x_0 x_1 x_2 x_3 x_4, \quad (4.17)$$

in  $\mathbb{CP}_{11111}^4$  with coordinates  $x_0, \dots, x_4$ . To obtain a polynomial system let us introduce the complex coordinate  $U = \nu + i u$  which relates to  $\psi$  as

$$U \simeq -\frac{5}{2\pi i} \log(5\psi), \quad (4.18)$$

up to corrections of  $\mathcal{O}(U^k e^{2\pi i U})$  that are exponentially suppressed in the large complex structure limit  $\text{Im}(U) \rightarrow \infty$ . This moduli space can be described by an approximately polynomial prepotential in the large complex structure limit

$$\mathcal{G} = \omega_0^2 \left( -\frac{5}{6} \left( \frac{\omega_1}{\omega_0} \right)^3 - \frac{11}{4} \left( \frac{\omega_1}{\omega_0} \right)^2 + \frac{25}{12} \frac{\omega_1}{\omega_0} - \frac{25\zeta(3)\chi}{2(2\pi i)^3} \right), \quad (4.19)$$

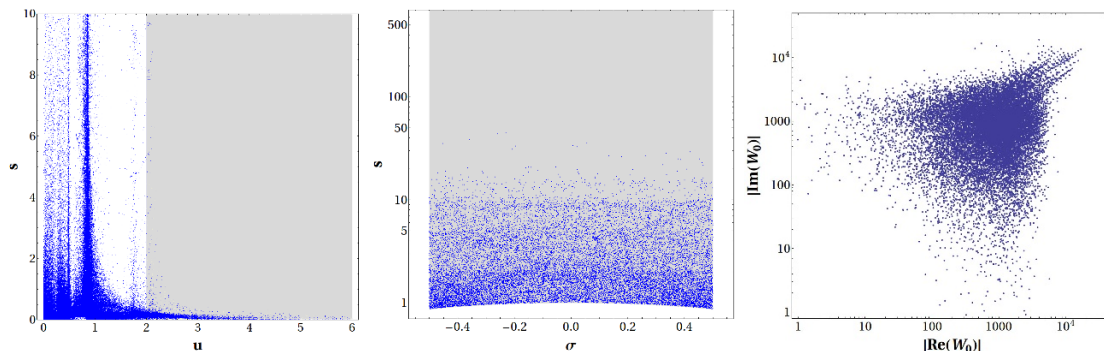
with  $\chi = 2(h^{1,1} - h^{2,1}) = 200$  and  $\omega_a$  and  $\mathcal{G}_b$  are the periods

$$\omega_a = \int_{A_a} \Omega, \quad \mathcal{G}_b = \int_{B^b} \Omega. \quad (4.20)$$

for  $a, b = 1, 2$  and  $\mathcal{G}_b = \partial_{\omega_b} \mathcal{G}$ .  $\omega_0$  can be interpreted as the normalization  $\Omega$  of the holomorphic three-form, such that the one physical variable is  $U = \omega_1/\omega_0$ . Eq. (4.19) is valid up to corrections  $\mathcal{O}(e^{2\pi i k U})$  that are exponentially small in the limit of the large complex structure. We define the large complex structure limit as  $\text{Im}(U) > 2$  such that these corrections to  $\mathcal{G}$  are smaller than  $10^{-3}\mathcal{G}$ .<sup>4</sup>

---

<sup>4</sup>Note that there is a conifold singularity at  $\psi^5 = 1$  that is excluded from the large complex structure limit  $\text{Im}(U) > 2$  according to eq. (4.18).



**Figure 4.** Left: distribution of *all* algebraic solutions to eq. (4.9) with  $s > 0$  and  $u > 0$  for the Kähler potential and superpotential (4.21) in *Sys3*. The gray shaded area indicates the space of physical solutions, i.e.  $s > 0$  and  $u > 2$ . Middle: distribution of the dilaton  $\tau = \sigma + is$ . Right: distribution of the superpotential  $W_0$ .

Together with the axio-dilaton,  $\tau$  this yields a two moduli example described by the Kähler potential and superpotential

$$\begin{aligned}
 K &= K_k - \log \left( -i \int_{X_3} \Omega(U) \wedge \bar{\Omega}(\bar{U}) \right) - \log(-i(\tau - \bar{\tau})) , \\
 &= -\log \left( i \sum_{a=1}^2 (\bar{\omega}_a \mathcal{G}_a - \omega_a \bar{\mathcal{G}}_a) \right) - \log(-i(\tau - \bar{\tau})) , \\
 W_0 &= \frac{1}{2\pi} \int_{X_3} (F_3 - \tau H_3) \wedge \Omega(U) , \\
 &= 2\pi \sum_{a=1}^2 [(f_{1a} - \tau h_{1a}) \mathcal{G}_a - (f_{2a} - \tau h_{2a}) U_a] ,
 \end{aligned} \tag{4.21}$$

where  $K_k$  is the Kähler potential of the Kähler moduli and setting  $\alpha' = 1$ .

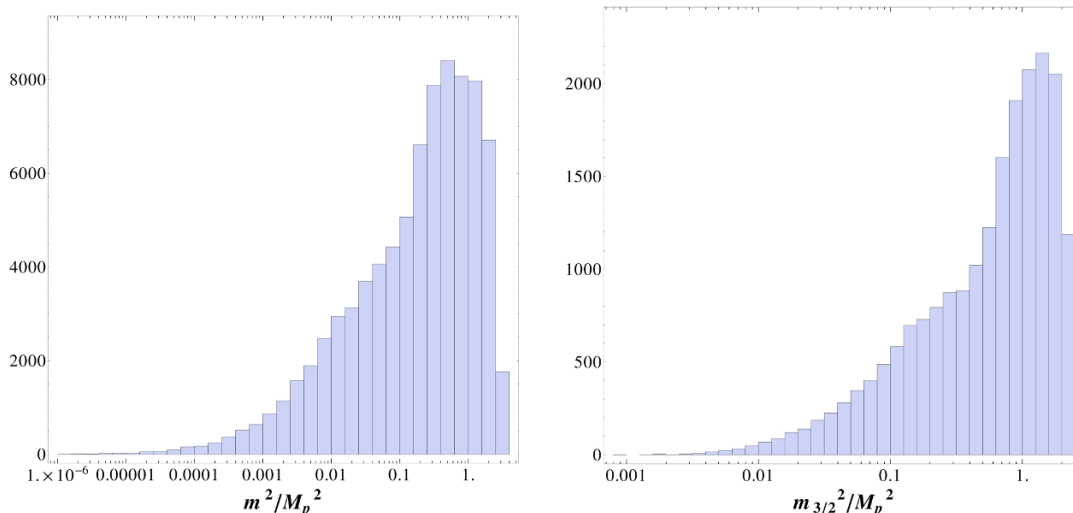
We solve eqs. (4.9) for (4.21) up to a maximal tadpole of  $L_{\max} = 625$ . This corresponds to 481,825 flux parameter points for which we find a total of 1,726,334 solutions, i.e. on average  $\sim 3.6$  solutions per parameter point. Only 20,280 are physical solutions, i.e.  $g_s^{-1} = s > 0$  and  $u > 2$ , see figure 4. If we were to solve it at each parameter-point from scratch, then it takes 72 minutes per parameter-point. This means that for total 481,825 parameter-points that we solved the system for, we would have taken  $481,825 \times 72/60 = 578,190$  hours. But instead, using Paramotopy, we solved all of them in only 3776 hours, i.e., around 26.4 human hours with a cluster of 144 processors. We could not obtain the CGB for this system in a reasonable time.

We can make use of the  $SL(2, \mathbb{Z})$  symmetry of IIB string theory, to transform each solution to the fundamental domain

$$-\frac{1}{2} \leq \text{Re}(\tau) \leq \frac{1}{2} \quad \text{and} \quad |\tau| > 1 , \tag{4.22}$$

via the successive transformations

$$\tau' = \tau + b, \quad W'_0 = W_0, \quad \text{and} \quad \tau' = -1/\tau, \quad W'_0 = W_0/\tau \tag{4.23}$$



**Figure 5.** For *Sys3* ((4.9) and (4.21)), the distribution of the masses  $m^2$  of the moduli  $t, \tau, s$  and  $\sigma$ , i.e. all eigenvalues of the Hessian of  $V$  (left) and the gravitino mass  $m_{3/2}^2$  (right) in units of  $M_p^2$  for  $\mathcal{V} = 100$ .

with  $b \in \mathbb{Z}$ . We show the distribution of the obtained values for  $\tau = \sigma + i s$  and the flux superpotential  $W_0$  in figure 4. We see that the strongly coupled region  $s = 1/g_s \sim 1$  is preferred and values of  $W_0 \sim \mathcal{O}(10^2 - 10^3)$  are preferred. The same qualitative behavior was observed in [81] for the manifold  $\mathbb{CP}_{11169}^4$ [18].

Finally, we give the masses  $m^2$  of the moduli  $t, \tau, s$  and  $\sigma$  and the gravitino mass  $m_{3/2}^2$  in figure 5. For this, we have to specify the value of the Kähler moduli Kähler potential in eq. (4.21). For a Calabi-Yau compactification, this is given as  $K_k = -2 \log \mathcal{V}$ , where  $\mathcal{V}$  is the volume modulus of the Calabi-Yau and its vacuum expectation value depends on the stabilization mechanism of the Kähler moduli. Here we choose  $\mathcal{V} = 100$  in string units for definiteness.

#### 4.4 Further applications: *Sys4*

As mentioned in the introduction, large systems with a multitude of parameters are ubiquitous. Thus, our methods above should be applicable to far more situations than finding the extrema of a PEL in the context of effective supersymmetric Lagrangians. In this subsection, as a parting example, let us see the power of the Paramotopy software applied to the geometry of Calabi-Yau manifolds.

For concreteness and continuing along the vein of *Sys3*, let us focus on the problem of finding the singular locus (or the absence thereof) given a family of quintics.<sup>5</sup>

Let us consider the family of quintic manifolds given by as a homogeneous hypersurface in  $\mathbb{CP}^4$  with coordinates  $x_0, \dots, x_4$ , and let

$$Q(x_0, \dots, x_4; a, b, c) = x_0^5 + x_1^5 + x_2^5 + x_3^5 + x_4^5 + ax_0x_1x_2x_3x_4 + bx_0^4x_1 + cx_1^3x_2x_3, \quad (4.24)$$

<sup>5</sup> Recently there has been much activity in study the database of so-called complete intersection Calabi-Yau (CICY) manifolds of which the quintic is the simplest example as well their smooth quotients. Checking smoothness for models with this database of 7890 Calabi-Yau threefolds and their descendents, for example, is a crucial step [85–87].

#sol	0	1–10	11–100	101–240
$P_0$	35,613	93,162	1,250	2,626
$P_1$	0	109,326	20,724	2,601
$P_2, P_3$	0	126,159	6,441	51
$P_4$	60	129,990	0	2,601

**Table 1.** Distributions of the number of solutions #sol for the five systems  $P_i$ .

where  $a, b, c$  are complex parameters. Of course, the most general family of quintics has 101 deformation parameters corresponding to  $h^{2,1}$  of the manifold, but this example suffices to show the power of the method. Now, for  $b = c = 0$ , we have the one most familiar to us, with  $a = -5$  being the conifold point in the complex structure moduli space. The singular locus of  $Q$  is given by the Jacobian ideal  $\langle \partial_{x_i} Q \rangle$  for  $i = 0, \dots, 4$ , excluding the origin and defined up to scaling of the projective coordinates.

It is more convenient to work over affine patches of  $\mathbb{CP}^4$  which can be rescaled to be  $P_i = \{x_i = 1\}$  for  $i = 0, \dots, 4$ . For each of the five patches, we compute the four remaining partial derivatives, the solution of which is then the system we need to analyze. That is, we need to perform **Paramotopy** on

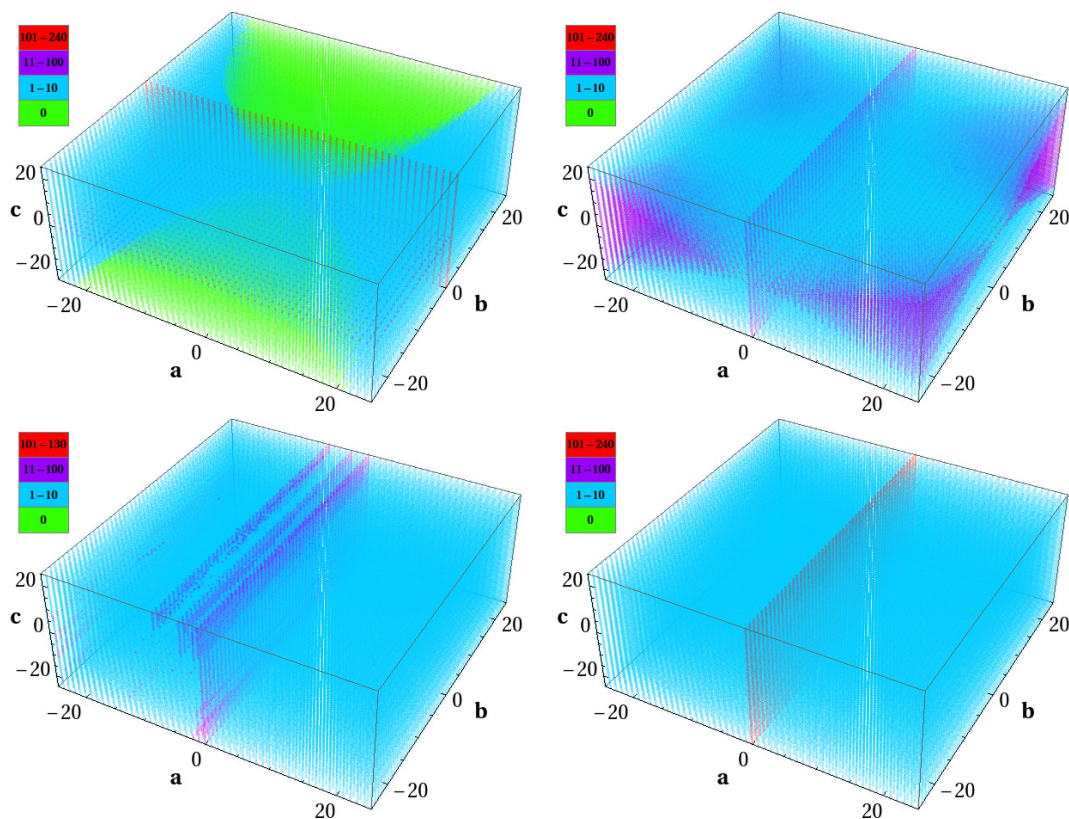
$$\partial_{x_{j \neq i}} Q(x_i = 1; a, b, c) = 0, \quad i = 0, \dots, 4. \tag{4.25}$$

All of the 5 systems are of the same size and we take  $a, b, c \in [-25, 25]$  in increments of one, i.e.  $51^3 = 132,651$  parameter points. **Bertini** takes from 144 to 223 seconds at a parameter-point if solved from scratch, hence it would have taken from 0.6057 to 0.938854 years for all 132651 parameter-points. **Paramotopy** solved each of these systems for all parameter-points in from 19.216 to 92.9002 hours (using 72 processors, it took 960.812 to 4645.01 seconds). Obtaining the CGB for this system is quite fast as well; however, obtaining the solutions for each parameter-point from the CGB takes the same amount of time as **Paramotopy**. Hence, we show the results only obtained from **Paramotopy** here. We show the number of solutions per parameter point indicated by the different colors in figure 6 and table 1. The solution space of the systems  $P_2$  and  $P_3$  is identical since eq. (4.24) is invariant under  $x_2 \leftrightarrow x_3$ .

## 5 Conclusion and outlook

Parametric systems of non-linear equations arise very naturally and frequently in theoretical physics. The parameters add one more hurdle in solving such systems in addition to the non-linearity of the equations as one has to solve the system many times. In the present paper, we have demonstrated that two of the sophisticated algebraic geometry methods can solve such systems very efficiently provided that the nonlinearity is polynomial-like.

The first method is the Comprehensive Gröbner basis (CGB) which is a symbolic method based on Gröbner bases. Given a (non-parametric) system of equations, we can systematically find another system of equations which has the same solution space as the original one and is easier to solve, called a Gröbner basis. If we have a parametric system



**Figure 6.** The number of solutions at each parameter point  $\{a, b, c\}$  for system  $P_0$  (top left),  $P_1$  (top right),  $P_2$  and  $P_3$  (bottom left) and  $P_4$  (bottom right) indicated by the different colors.

of equations, then we can compute the CGB which is a Gröbner basis for all values of the parameters including their special values. To obtain the solutions of the system at a specific parameter-point, one then needs to input the values of the parameters in the CGB and solve the system using the traditional methods. However, the algorithmic complexity to compute the CGB is the same, or in many cases even worse, as the Gröbner basis method, i.e., the memory required by the algorithm blows up exponentially with increasing number of variables, equations, monomials, and degree. Moreover, the related algorithm is highly sequential and unparallelizable. Hence, though the CGB method is extremely useful in the cases where we can finish the computation, but for many physical systems the method often falls short.

The second method we described in this paper is called coefficient-parameter polynomial continuation. Here, one first solves the given parametric system of polynomial equations at a generic parameter-point because it is shown that the number of solutions at such a parameter-point is an upper bound on the number of solutions at any other parameter-points. Then, one tracks solution-paths from the solutions at the generic parameter-point to the ones at which the system needs to be solved using the numerical polynomial homotopy continuation method. This method is highly parallelizable, and, hence, we can solve the system at thousands of parameter-points in a short time on a computer cluster. We

System	Number of parameter points	single-processor	CPPC
<i>Sys1</i>	100,150	5.731 yr	55 h
<i>Sys2</i>	100,672	7.682 yr	125 h
<i>Sys3</i>	481,825	66.003 yr	3776 h
<i>Sys4</i>	132,651	0.6 – 0.9 yr	19.2 – 92.9 h

**Table 2.** Number of parameter points and computation time for the examples under consideration.

use a package called *Paramotopy* which is not only a highly efficient implementation of the related algorithm, but also does data and memory management effectively. These features allow for a huge number of parameter-points to be solved simultaneously.

We have used these two methods on various examples arising from computing the extrema of potential energy landscapes, especially in the context of moduli stabilization in string phenomenology. However, obtaining the solutions for each parameter-point from the CGB (for the cases it is possible to obtain the CGB) takes the same amount of time as *Paramotopy*; hence, we show the results only obtained from CPPC in this paper.

The examples are given as a 4D  $\mathcal{N} = 1$  effective supergravity scalar potential, defined by a Kähler potential  $K$  and a superpotential  $W$ . Our first two examples *Sys1* and *Sys2* are toy models of one and two complex scalar fields, respectively.

We looked for stationary points of these scalar potentials for  $\sim 10^5$  parameter points, finding half of the stationary points are vacua in the case of *Sys1* and finding no vacua in the case of *Sys2*, i.e., at least one tachyonic direction always exists. Finally, we looked at flux compactifications on a realistic Calabi-Yau, the quintic in the case of two moduli. We look for flux vacua for  $\sim 5 \cdot 10^5$  flux configurations, which corresponds to a maximal D3 tadpole of  $L = 625$ . Our results indicate preferences of strongly coupled vacua  $g_s \gtrsim 1$  and values of the superpotential  $W_0 \sim \mathcal{O}(10^2 - 10^3)$ .

The applicability of the methods described here is, of course, much wider and is intended for any large parametric polynomial systems, with the parameters appearing explicitly as coefficients. Indeed, these methods should provide an extremely powerful tool for many areas in theoretical physics such as potential energy landscape, statistical mechanics, particle phenomenology, string phenomenology, and non-linear dynamics. As an illustration, we have applied CPPC to the geometry of Calabi-Yau manifolds in the case of *Sys4*.

If we were to solve *Sys1*, *Sys2*, the quintic, and the Calabi-Yau systems from scratch for each parameter-point using the numerical homotopy continuation method, a single processor machine would have taken 5.731 years, 7.682 years, 66.003 years, and 0.6057-0.938854 years, respectively. But using coefficient-parameter polynomial continuation, we solved them in 55 hours, 125 hours, 3776 hours and 19.216-92.9002 hours, respectively. This is a drastic reduction of the computational efforts, see also table 2.

We emphasize that the two methods are based on complex algebraic geometry which means that the variables are first considered to be complex, and then only purely real solutions are retained, if desired. Because in many applications we only have real parameters

and only real solutions are physically important, a natural question to ask is if any method exists which can deal with systems with both parameters and variables defined over reals. Indeed, a more recent method exists from symbolic real algebraic geometry, based on the discriminant varieties [52, 53], which treats both variables and parameters as reals from the beginning and gives a different set of information than the methods described in this paper. This certainly constitutes a direction worthy of pursuit. On the numerical side, little progress exists in this direction, and so far methods exist to extract only limited information out for the whole parameter space. For example, in ref. [88], a numerical method is proposed which can tell us the maximum and minimum number of real solutions over the whole parameter-space. It is our hope that the applications in this paper will also become a motivation to develop such methods and making them more for real-life applications. One can also *certify* if a given solution is indeed in the quadratic convergence region of the system using Smale's  $\alpha$ -theory which is used in the potential energy landscape areas in [89]. In future, we plan to combine CPPC method and the certification method to achieve the 'exact' solutions of a given system.

## Acknowledgments

DM would like to thank the U.S. Department of Energy for their support under contract no. DE-FG02-85ER40237; to Tianran Chen and Tim McCoy for their helpful remarks on this work; and to the entire Bertini, PHCpack and HOM4PS2 groups for their continuous support in this project. A major part of this paper is based on DM's PhD thesis. MN wishes to thank the NSF for their partial support under grant DMS-1025564. YHH would like to thank the Science and Technology Facilities Council, UK, for an Advanced Fellowship and grant ST/J00037X/1, the Chinese Ministry of Education, for a Chang-Jiang Chair Professorship at NanKai University, the U.S. National Science Foundation for grant CCF-1048082, as well as City University, London and Merton College, Oxford, for their enduring support. MR would like to thank Alexander Westphal for helpful discussions.

## References

- [1] D. Wales, *Energy landscapes: applications to clusters, biomolecules and glasses*, Cambridge Molecular Science, Cambridge University Press, Cambridge U.K. (2004).
- [2] D.J. Wales, *The energy landscape as a unifying theme in molecular science*, *Phil. Trans. Roy. Soc. A* **363** (2005) 357.
- [3] D.J. Wales, *Energy landscapes: calculating pathways and rates*, *Int. Rev. Phys. Chem.* **25** (2006) 237.
- [4] D.J. Wales and T.V. Bogdan, *Potential energy and free energy landscapes*, *J. Phys. Chem. B* **110** (2006) 20765.
- [5] T.F. Middleton and D.J. Wales, *Energy landscapes of model glasses. II. Results for constant pressure*, *J. Chem. Phys.* **118** (2003) 4583.
- [6] D.J. Wales and J.P.K. Doye, *Dynamics and thermodynamics of supercooled liquids and glasses from a model energy landscape*, *Phys. Rev. B* **63** (2001) 214204.



- [7] K. Binder and A. Young, *Spin glasses: experimental facts, theoretical concepts, and open questions*, *Rev. Mod. Phys.* **58** (1986) 801.
- [8] M. Kastner, *Phase transitions and configuration space topology*, *Rev. Mod. Phys.* **80** (2008) 167 [[INSPIRE](#)].
- [9] L.E. Ibanez and A.M. Uranga, *String theory and particle physics: an introduction to string phenomenology*, *Phys. Today* **66** (2013) 51.
- [10] K. Dasgupta, G. Rajesh and S. Sethi, *M theory, orientifolds and G-flux*, *JHEP* **08** (1999) 023 [[hep-th/9908088](#)] [[INSPIRE](#)].
- [11] J. McOrist and S. Sethi, *M-theory and type IIA flux compactifications*, *JHEP* **12** (2012) 122 [[arXiv:1208.0261](#)] [[INSPIRE](#)].
- [12] S.B. Giddings, S. Kachru and J. Polchinski, *Hierarchies from fluxes in string compactifications*, *Phys. Rev. D* **66** (2002) 106006 [[hep-th/0105097](#)] [[INSPIRE](#)].
- [13] M.R. Douglas and S. Kachru, *Flux compactification*, *Rev. Mod. Phys.* **79** (2007) 733 [[hep-th/0610102](#)] [[INSPIRE](#)].
- [14] D. Marsh, L. McAllister and T. Wrase, *The wasteland of random supergravities*, *JHEP* **03** (2012) 102 [[arXiv:1112.3034](#)] [[INSPIRE](#)].
- [15] M. Graña, *Flux compactifications in string theory: a comprehensive review*, *Phys. Rept.* **423** (2006) 91 [[hep-th/0509003](#)] [[INSPIRE](#)].
- [16] R. Blumenhagen, B. Körs, D. Lüst and S. Stieberger, *Four-dimensional string compactifications with D-branes, orientifolds and fluxes*, *Phys. Rept.* **445** (2007) 1 [[hep-th/0610327](#)] [[INSPIRE](#)].
- [17] D.J. Wales, *Basins of attraction for stationary-points on a potential-energy surface*, *J. Chem. Soc. Faraday Trans.* **88** (1992) 653.
- [18] D.J. Wales, *Locating stationary-points for clusters in cartesian coordinates*, *J. Chem. Soc. Faraday Trans.* **89** (1993) 1305.
- [19] K. Broderix, K. Bhattacharya, A. Cavagna, A. Zippelius and I. Giardina, *Energy landscape of a lennard-jones liquid: statistics of stationary points*, *Phys. Rev. Lett.* **85** (2000) 5360 [[cond-mat/0007258](#)].
- [20] J.P.K. Doye and D.J. Wales, *Saddle points and dynamics of Lennard-Jones clusters, solids and supercooled liquids*, *J. Chem. Phys.* **116** (2002) 3777 [[cond-mat/0108310](#)].
- [21] D. Wales and J. Doye, *Stationary points and dynamics in high-dimensional systems*, *J. Chem. Phys.* **119** (2003) 12409.
- [22] J. Gray, Y.-H. He, A. Ilderton and A. Lukas, *A new method for finding vacua in string phenomenology*, *JHEP* **07** (2007) 023 [[hep-th/0703249](#)] [[INSPIRE](#)].
- [23] Y.-H. He et al., *Computational algebraic geometry in string and gauge theory*, Special Issue, Advances in High Energy Physics, Hindawi publishing, Egypt (2012).
- [24] J. Gray, Y.-H. He, V. Jejjala and B.D. Nelson, *Exploring the vacuum geometry of  $N = 1$  gauge theories*, *Nucl. Phys. B* **750** (2006) 1 [[hep-th/0604208](#)] [[INSPIRE](#)].
- [25] J. Gray, A. Hanany, Y.-H. He, V. Jejjala and N. Mekareeya, *SQCD: a geometric apercu*, *JHEP* **05** (2008) 099 [[arXiv:0803.4257](#)] [[INSPIRE](#)].

- [26] J. Gray, *A simple introduction to Gröbner basis methods in string phenomenology*, *Adv. High Energy Phys.* **2011** (2011) 217035 [[arXiv:0901.1662](#)] [[INSPIRE](#)].
- [27] J. Gray, Y.-H. He and A. Lukas, *Algorithmic algebraic geometry and flux vacua*, *JHEP* **09** (2006) 031 [[hep-th/0606122](#)] [[INSPIRE](#)].
- [28] J. Gray, Y.-H. He, A. Ilderton and A. Lukas, *STRINGVACUA: a Mathematica package for studying vacuum configurations in string phenomenology*, *Comput. Phys. Commun.* **180** (2009) 107 [[arXiv:0801.1508](#)] [[INSPIRE](#)].
- [29] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, *Singular 3-1-3 — A computer algebra system for polynomial computations*, <http://www.singular.uni-kl.de>.
- [30] D. Mehta, *Lattice vs. continuum: Landau gauge fixing and 't Hooft-Polyakov monopoles*, Ph.D. Thesis, University of Adelaide, Adelaide, Australia (2009).
- [31] D. Mehta, A. Sternbeck, L. von Smekal and A.G. Williams, *Lattice Landau Gauge and Algebraic Geometry*, *PoS(QCD-TNT09)025* [[arXiv:0912.0450](#)] [[INSPIRE](#)].
- [32] C. Hughes, D. Mehta and J.-I. Skullerud, *Enumerating Gribov copies on the lattice*, *Annals Phys.* **331** (2013) 188 [[arXiv:1203.4847](#)] [[INSPIRE](#)].
- [33] D. Mehta, *Finding all the stationary points of a potential energy landscape via numerical polynomial homotopy continuation method*, *Phys. Rev. E* **84** (2011) 025702 [[arXiv:1104.5497](#)] [[INSPIRE](#)].
- [34] M. Kastner and D. Mehta, *Phase transitions detached from stationary points of the energy landscape*, *Phys. Rev. Lett.* **107** (2011) 160602 [[arXiv:1108.2345](#)] [[INSPIRE](#)].
- [35] D. Mehta, J.D. Hauenstein and M. Kastner, *Energy landscape analysis of the two-dimensional nearest-neighbor  $\phi^4$  model*, *Phys. Rev. E* **85** (2012) 061103 [[arXiv:1202.3320](#)] [[INSPIRE](#)].
- [36] R. Nerattini, M. Kastner, D. Mehta and L. Casetti, *Exploring the energy landscape of XY models*, *Physical Review E* **87**, **032140** (2013) [[arXiv:1211.4800](#)] [[INSPIRE](#)].
- [37] D. Mehta, D.A. Stariolo and M. Kastner, *Energy landscape of the finite-size mean-field 3-spin spherical model*, to be published in *Phys. Rev. E*, [arXiv:1303.1520](#) [[INSPIRE](#)].
- [38] M. Maniatis and D. Mehta, *Minimizing Higgs potentials via numerical polynomial homotopy continuation*, *Eur. Phys. J. Plus* **127** (2012) 91 [[arXiv:1203.0409](#)] [[INSPIRE](#)].
- [39] J. Camargo-Molina, B. O'Leary, W. Porod and F. Staub, *The stability of R-parity in supersymmetric models extended by  $U(1)_{B-L}$* , [arXiv:1212.4146](#) [[INSPIRE](#)].
- [40] D. Mehta, *Numerical polynomial homotopy continuation method and string vacua*, *Adv. High Energy Phys.* **2011** (2011) 263937 [[arXiv:1108.1201](#)] [[INSPIRE](#)].
- [41] D. Mehta, Y.-H. He and J.D. Hauenstein, *Numerical algebraic geometry: a new perspective on string and gauge theories*, *JHEP* **07** (2012) 018 [[arXiv:1203.4235](#)] [[INSPIRE](#)].
- [42] J. Hauenstein, Y.-H. He and D. Mehta, *Numerical analyses on moduli space of vacua*, [arXiv:1210.6038](#) [[INSPIRE](#)].
- [43] B. Greene, D. Kagan, A. Masoumi, E.J. Weinberg and X. Xiao, *Tumbling through a landscape: evidence of instabilities in high-dimensional moduli spaces*, [arXiv:1303.4428](#) [[INSPIRE](#)].
- [44] D.J. Wales, *A microscopic basis for the global appearance of energy landscapes*, *Science* **293** (2001) 2067.

- [45] D.J. Wales, *Highlights: Energy landscapes of clusters bound by short-ranged potentials*, *Chem. Phys. Chem* **11** (2010) 2491.
- [46] D.J. Wales, J.P.K. Doye and D.J. Wales, *Energy landscapes of colloidal clusters: Thermodynamics and rearrangement mechanisms*, *Nanoscale* **4** (2012) 1085.
- [47] D.J. Wales, *Decoding the energy landscape: Extracting structure, dynamics and thermodynamics*, *Phil. Trans. Roy. Soc. A* **370** (2012) 2877.
- [48] L. von Smekal, D. Mehta, A. Sternbeck and A.G. Williams, *Modified lattice Landau gauge*, *PoS(LATTICE 2007)382*.
- [49] L. von Smekal, A. Jorkowski, D. Mehta and A. Sternbeck, *Lattice Landau gauge via stereographic projection*, *PoS(CONFINEMENTS)048* [[arXiv:0812.2992](#)] [[INSPIRE](#)].
- [50] A.P. Morgan and A.J. Sommese, *Coefficient-parameter polynomial continuation*, *Appl. Math. Comput.* **29** (1989) 123.
- [51] D.J. Bates, D.A. Brake and M.E. Niemerg., *Paramotopy: parameter homotopies in parallel*, in preparation.
- [52] W. Hanan, D. Mehta, G. Moroz and S. Pouryahya, *Stability and bifurcation analysis of coupled Fitzhugh-Nagumo oscillators*, extended abstract published in the proceedings of the *Joint Conference of ASCM2009 and MACIS2009*, Japan (2009), [arXiv:1001.5420](#).
- [53] E.A. Hernandez-Vargas, D. Mehta and R.H. Middleton, *Towards modeling HIV long term behavior*, in the proceedings of the 18<sup>th</sup> *IFAC World Congress*, August 28–September 2, Milano, Italy (2011), [arXiv:1105.2823](#).
- [54] D.A. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, Undergraduate Texts in Mathematics, Springer, U.S.A. (2007).
- [55] J.C. Faugère, *A new efficient algorithm for computing Groebner bases ( $F_4$ )*, *J. Pure Appl. Alg.* **139** (1999) 61.
- [56] J.C. Faugère, *A new efficient algorithm for computing Gröbner bases without reduction to zero ( $F_5$ )*, in the proceedings of the *International symposium on Symbolic and algebraic computation (ISSAC02)*, July 7–10, New York, U.S.A. (2002).
- [57] V.P. Gerdt, *Involutive algorithms for computing Groebner bases*, [math/0501111](#).
- [58] CoCoATeam, *CoCoA: a system for doing Computations in Commutative Algebra*, <http://cocoa.dima.unige.it>.
- [59] D.R. Grayson and M.E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, <http://www.math.uiuc.edu/Macaulay2/>.
- [60] W. Bosma, J. Cannon and C. Playoust, *The Magma algebra system I: the user language*, *J. Symb. Comput.* **24** (1997) 235.
- [61] V. Weispfenning, *Comprehensive Gröbner bases*, *J. Symb. Comp.* **14** (1992) 1.
- [62] D.A. Cox, J. Little and D. O’Shea, *Using algebraic geometry*, Springer, U.S.A. (1998).
- [63] A.J. Sommese and C.W. Wampler, *The numerical solution of systems of polynomials arising in engineering and science*, World Scientific Publishing Company, Singapore (2005).
- [64] A. Montes, *A new algorithm for discussing Gröbner bases with parameters*, *J. Symb. Comput.* **33** (2002) 183.

- [65] A. Suzuki and Y. Sato, *An alternative approach to comprehensive Gröbner bases*, *J. Symb. Comput.* **36** (2003) 649.
- [66] D. Kapur, Y. Sun and D. Wang, *An efficient method for computing comprehensive Gröbner bases*, *J. Symb. Comput.* **52** (2013) 124.
- [67] T.Y. Li, *Solving polynomial systems by the homotopy continuation method*, in *Handbook of Numerical Analysis XI*, P.G. Ciarlet and J.L. Lion eds., Gulf Professional Publishing, U.S.A. (2003).
- [68] D.J. Bates, J.D. Hauenstein, A.J. Sommese and C.W. Wampler, *Bertini: software for numerical algebraic geometry*, <http://www.nd.edu/~sommese/bertini>.
- [69] J. Verschelde, *Algorithm 795: Phcpack: a general-purpose solver for polynomial systems by homotopy continuation*, *ACM Trans. Math. Soft.* **25** (1999) 251.
- [70] A.P. Morgan, A.J. Sommese and L.T. Watson, *Finding all isolated solutions to polynomial systems using hompack*, *ACM Trans. Math. Softw.* **15** (1989) 93.
- [71] T. Gao, T.Y. Li and M. Wu, *Algorithm 846: Mixedvol: a software package for mixed-volume computation*, *ACM Trans. Math. Softw.* **31** (2005) 555.
- [72] T.L. Lee, T.Y. Li and C.H. Tsai, *Hom4ps-2.0, a software package for solving polynomial systems by the polyhedral homotopy continuation method*, *Computing* **83** (2008) 109.
- [73] D.N. Bernstein, *The number of roots of a system of equations*, *Funkts. Anal. Pril.* **9** (1975) 1.
- [74] A.G. Khovanski, *Newton polyhedra and the genus of complete intersections*, *Funkts. Anal. Pril.* **12** (1978) 51.
- [75] A. G. Kushnirenko, *Newton polytopes and the bezout theorem*, *Funkts. Anal. Pril.* **10** (1976) 82.
- [76] T. Li, T. Sauer and J. Yorke, *The cheater's homotopy: an efficient procedure for solving systems of polynomial equations*, *SIAM J. Numer. Anal.* **26** (1989) 1241.
- [77] J. Wess and J. Bagger, *Supersymmetry and supergravity*, Princeton University Press, Princeton U.S.A. (1992).
- [78] E. Cremmer, S. Ferrara, C. Kounnas and D.V. Nanopoulos, *Naturally vanishing cosmological constant in  $N = 1$  supergravity*, *Phys. Lett.* **B 133** (1983) 61 [INSPIRE].
- [79] J.R. Ellis, A. Lahanas, D.V. Nanopoulos and K. Tamvakis, *No-scale supersymmetric standard model*, *Phys. Lett.* **B 134** (1984) 429 [INSPIRE].
- [80] F. Denef and M.R. Douglas, *Distributions of flux vacua*, *JHEP* **05** (2004) 072 [[hep-th/0404116](http://arxiv.org/abs/hep-th/0404116)] [INSPIRE].
- [81] D. Martinez-Pedrerá, D. Mehta, M. Rummel and A. Westphal, *Finding all flux vacua in an explicit example*, [arXiv:1212.4530](http://arxiv.org/abs/1212.4530) [INSPIRE].
- [82] M. Rummel and A. Westphal, *A sufficient condition for de Sitter vacua in type IIB string theory*, *JHEP* **01** (2012) 020 [[arXiv:1107.2115](http://arxiv.org/abs/1107.2115)] [INSPIRE].
- [83] J. Louis, M. Rummel, R. Valandro and A. Westphal, *Building an explicit de Sitter*, *JHEP* **10** (2012) 163 [[arXiv:1208.3208](http://arxiv.org/abs/1208.3208)] [INSPIRE].
- [84] P. Candelas, X.C. De La Ossa, P.S. Green and L. Parkes, *A pair of Calabi-Yau manifolds as an exactly soluble superconformal theory*, *Nucl. Phys.* **B 359** (1991) 21 [INSPIRE].

- [85] P. Candelas, A. Dale, C. Lütken and R. Schimmrigk, *Complete intersection Calabi-Yau manifolds*, *Nucl. Phys. B* **298** (1988) 493 [INSPIRE].
- [86] L.B. Anderson, Y.-H. He and A. Lukas, *Heterotic compactification, an algorithmic approach*, *JHEP* **07** (2007) 049 [[hep-th/0702210](http://arxiv.org/abs/hep-th/0702210)] [INSPIRE].
- [87] V. Braun, *On free quotients of complete intersection Calabi-Yau manifolds*, *JHEP* **04** (2011) 005 [[arXiv:1003.3235](http://arxiv.org/abs/1003.3235)] [INSPIRE].
- [88] Z.A. Griffin and J.D. Hauenstein, *Real solutions to systems of polynomial equations and parameter continuation*,  
<http://arxiv.org/abs/http://www4.ncsu.edu/~jdhauens/preprints/ghRealSolving.pdf>
- [89] D. Mehta, J.D. Hauenstein and D.J. Wales, *Certifying the potential energy landscape*, *J. Chem. Phys.* **138** (2013) 171101 [[arXiv:1302.6265](http://arxiv.org/abs/1302.6265)] [INSPIRE].